

## 53 sensore di corrente – electric meter (some notes at section end)



Quando si pensa ad un misuratore di corrente si pensa ad un amperometro mediante il quale misurare l'intensità della corrente erogata e, tramite un banale calcolo ( $\text{watt} = \text{volt} * \text{ampere}$ ), determinare il consumo in watt.

Esiste però un sistema, più pratico di quanto non lo sia un amperometro, in grado di fornire indicazioni sul consumo mediante la misurazione della corrente indotta in un bobina da uno dei due cavi in cui scorre la corrente che si intende misurare.

Il sensore TA12-100 (in figura), utilizzando questo secondo sistema, è in grado di misurare la corrente che sta transitando in un filo, fino ad un valore massimo di 5 ampere e quindi, considerando una tensione di 220 volt, consumi massimi di poco

superiori ai 1000 watt.

L'utilizzo di questo sensore è piuttosto semplice poiché è sufficiente far passare nel foro centrale della bobina uno dei due fili in cui transita la corrente che si vuole misurare (un solo filo e non tutti e due, poiché se fossero insieme annullerebbero ogni possibilità di misurazione).

Con un po' di approssimazione possiamo dire che il sensore è un trasformatore con rapporto 1000:1 e cioè che è composto da 1000 spire che vengono eccitate dalla corrente che transita in una sola spira (che sarebbe poi il filo di cui si vuole misurare la corrente, e che si fa passare nel foro al centro della bobina).

La debolissima corrente generata dal circuito secondario (quello con le 1000 spire) viene intercettata (e annullata) da una resistenza da 200 ohm posta alla base di detto circuito. A questo punto Arduino rileva, mediante una porta analogica, la caduta di tensione presente ai due capi della resistenza (e quindi la tensione prodotta dal circuito secondario).

Applicando la legge di ohm ( $\text{Intensità} = \text{Tensione} / \text{Resistenza}$ ) Arduino calcola gli ampere (anzi i milliampere) della corrente prodotta dal secondario e da questi, tramite il fattore di scala 0,707 (un valore costante per le onde sinusoidali), arriva al "valore efficace" o anche "RMS".

Detto valore, moltiplicato per 1000 (la proporzione è di 1000 spire contro una) rappresenta di fatto il numero degli ampere in transito nel cavo in misurazione.

Un discorso complesso e certamente impreciso, ma che dovrebbe aver reso l'idea di come funzioni questo interessante sensore.

**Attenzione:** Nel caso si volesse replicare questo esercizio si raccomanda la massima attenzione nell'utilizzo del sistema di cui si vuole misurare l'assorbimento, poiché in detto sistema circolano correnti ad alta tensione.

Prima di compilare il programma bisogna, scaricare ed installare, se non già fatto:

- La libreria di gestione del display a cristalli liquidi, reperibile [qui](#).

Per installare una libreria:

- Scaricarla mantenendola in formato compresso
- Andare in IDE-> sketch-> importa libreria-> add library->

## Arduino: Sensore di corrente – Current sensor

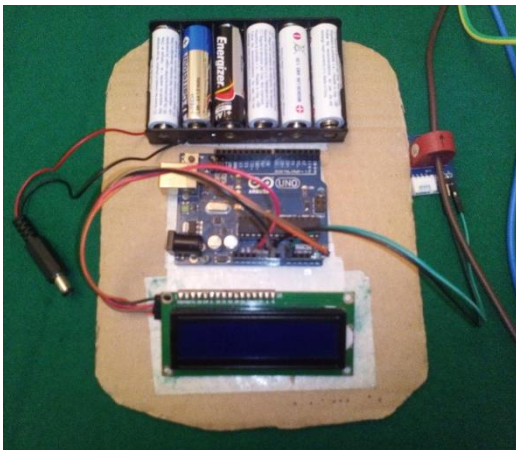
- Individuare la cartella di download->
- Fare doppio click sulla libreria da installare
- Verificarne la presenza in IDE-> sketch-> includes Library-> Contributed library

**Nota:** Questo esercizio e questa nota sono parte di una serie che vede protagonisti Arduino ed alcuni dei componenti ad esso collegabili. Per la maggior parte degli esercizi e' anche disponibile un filmato su youtube:

- [Esercizi facenti parte della raccolta](#)
- [Filmati presenti su youtube](#)
- [Informazioni su arduino e sui componenti collegabili \(PDF scaricato nell'area di download\)](#)
- [Breve manuale di programmazione \(PDF scaricato nell'area di download\)](#)

Per eventuali chiarimenti o suggerimenti sul contenuto di questa scheda scrivere a [giocarduino@libero.it](mailto:giocarduino@libero.it)

**Here some notes about this project, translated by google translator**



When we think of an electric meter, we think of an ammeter by which to measure the current intensity and, using a basic formula ( $\text{watt} = \text{volt} * \text{ampere}$ ), determine the consumption.

There is, however, a more practical system, able to provide indications on consumption by measuring the current induced in a coil by one of the two cables in which flows current (AC) to be measured.

The sensor TA12-100 (in figure), using this second system, is able to measure a current flowing through a wire, up to 5 amps, and therefore, considering a voltage of 220 volts, just over 1000 watts as maximum.

Use of this sensor is rather simple since it is sufficient to pass in the coil central hole one of the two wires in which flows current you want to measure (one wire only and not both, because if they are together they would cancel any possibility of measurement).

With a little approximation we can say that the sensor is a transformer with a ratio of 1000:1. A secondary composed by 1000 whorls excited by the current that passes in a primary composed by only one turn, which would be the wire passing through the hole in the middle of the coil.

The very weak current generated by the secondary circuit (the one with the 1000 turns) is intercepted by a 200 ohm resistor placed at the base of said circuit. At this point, Arduino detects, through an analogue port, the voltage drop present at the two ends of the resistor and therefore the voltage produced by the secondary circuit.

Applying the ohm law ( $\text{Intensity} = \text{Voltage} / \text{Resistance}$ ) Arduino calculates amperes (or rather the milliamps) produced by the secondary and from these, through the scale factor 0.707 (a constant value for sinusoidal waves), obtains the "effective value" or "RMS".

This value, multiplied by 1000 (the proportion is 1000 whorls against one) represents in fact the number of amps absorbed by device we are measuring.

A complex and certainly inaccurate speech, but that should make idea how this interesting sensor works.

**Warning:** If you want to replicate this project, we recommend the utmost care in connecting device you want to measure, since high voltage circulating.

Before proceeding to program compilation must be installed, if not already done, the library:

- LiquidCrystal\_I2C.h found in: <https://github.com/fmalpartida/New-LiquidCrystal>

For library installation, see process shown in previous projects, and summarized in:

- library download in compressed form;
- Installation via IDE-> sketch-> includes Library-> add .zip library
- After installation please verify the library. It must be present in IDE-> sketch-> includes Library-> Contributed library

**Note:** This project and this note is part of a series that sees, as main characters, Arduino and some of connectable components. For most projects there is also a video on youtube.

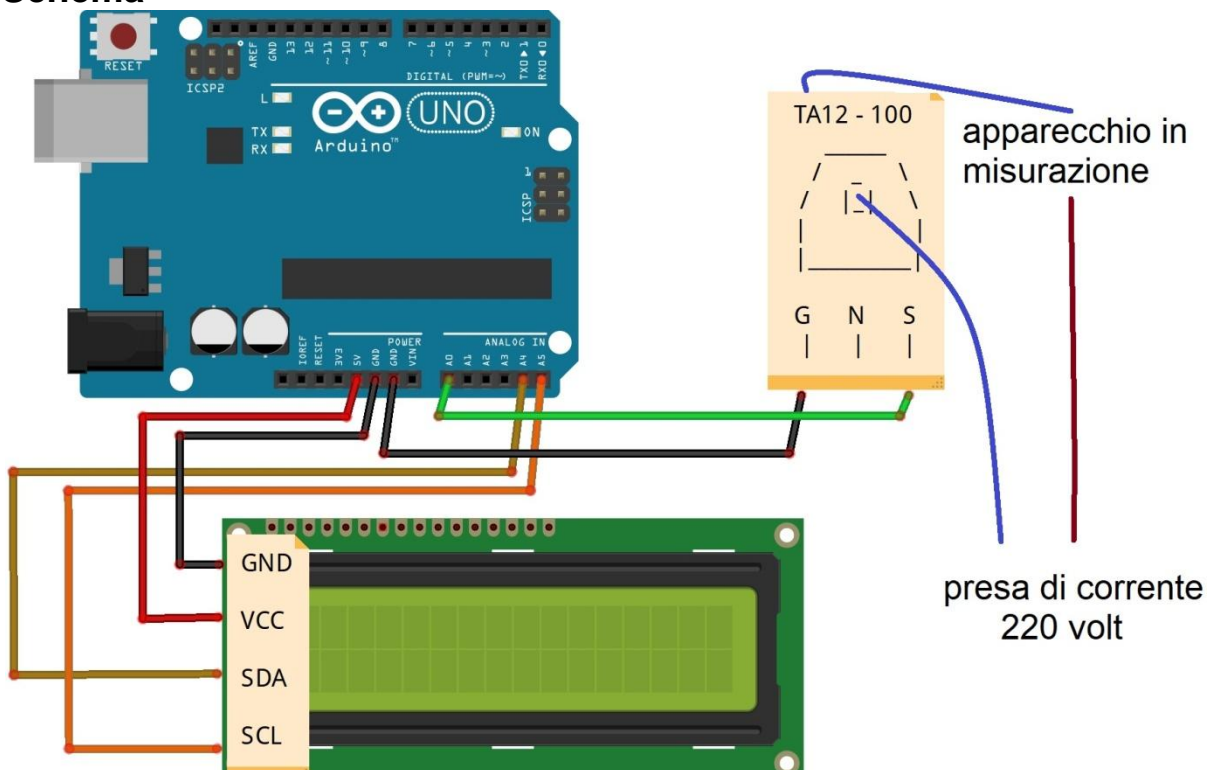
- [Projects collection](#)
- [Movies on youtube](#)
- [About Arduino and components \(italian; pdf will be downloaded in your download area\)](#)
- [Quick programming guide \(almost english; pdf will be downloaded in your download area\)](#)

For any questions or suggestions about this note (and on its english translation), please write to [giocarduino@libero.it](mailto:giocarduino@libero.it) (simple words and short sentences, please)

## Materiali

- Un sensore modello TA12-100
- Un display lcd con driver I2C

## Schema



## Programma

Download del programma tramite [Dropbox](#)  
Program download via [Dropbox](#)

```

/* Attenzione: facendo il copia/incolla dal PDF all'IDE si perde la formattazione del testo. Per
 * rendere piu' facilmente leggibile il programma e' opportuno formattarlo subito dopo il
 * trasferimento nell'IDE, premendo CTRL+T.
 * In questo esercizio si utilizza un sensore di corrente per rilevare gli ampere e calcolare i
 * watt in erogazione su di una utenza elettrica. Il sensore e' in pratica un trasformatore il
 * cui circuito primario e' rappresentato da uno dei due fili in cui transita la corrente, mentre il
 * secondario e' composto da 1000 spire. La caduta di tensione misurata ai capi di una resistenza da
 * 200 ohm che delimita il circuito secondario, viene gestita da arduino che, applicando La legge di
 * ohm e rettificando opportunamente il risultato, ottiene il "valore efficace" (e cioe'
 * l'intensita' della corrente in transito sul filo primario) e di conseguenza, indicazioni sulla
 * potenza assorbita dall'utenza in esame.
-----
 * Warning: cut&paste from PDF to IDE loses formatting. To restore it press CTRL + T.
 * In this project, a current sensor is used to detect amperes and calculate watts consumed by an
 * electrical device. Practically the sensor is a transformer whose primary circuit is represented
 * by one of the two wires powering the device, while the secondary one is made by a 1000 turns
 * coil. The voltage drop measured at the ends of a 200 ohm resistor that delimits the secondary
 * circuit, is managed by Arduino which, applying the ohms law and appropriately rectifying
 * result, obtains the "effective value" (ie the current intensity passing through the primary wire)
 * and, consequently, indications on power absorbed by device.
 */

# define sensore A0                // il sensore e' collegato alla porta analogica 0
#include <LiquidCrystal_I2C.h>      // libreria di gestione del display lcd
//..... addr, en,rw,rs,d4,d5,d6,d7,bl,blpol
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); // tipologia del display
// LiquidCrystal_I2C lcd(0x3F, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
// ^ indirizzo alternativo, da attivare al posto della precedente istruzione se lcd non funziona
// ^^ Alternative address, to activate in place of previous instruction if LCD doesn't work

int  statosensore      = 0;        // zona di memorizzazione del valore fornito dal sensore
int  picco             = 0;        // zona di memorizzazione del picco fornito dal sensore
float tensione         = 0;        // tensione di picco, misurata ai capi della resistenza
float intensita        = 0;        // intensita alla base della resistenza,calcolata con legge di ohm
float intensitarms     = 0;        // intensita' efficace, sempre alla base della resistenza
float ampere           = 0;        // ampere in erogazione sull'utenza
int  watt              = 0;        // watt istantanei
float momentocorrente  = 0;        // momento di rilevamento
float momentoesposizione = 0;     // momento in cui e' stata fatta l'ultima esposizione dei valori
//
//
// ***** routine di rilevazione del picco (di tensione) ai capi della resistenza *****
//
void rilevapicco ()
{
    statosensore = analogRead (sensore);
    if (statosensore > picco)
        picco = statosensore;      // memorizza sempre il valore piu' alto
}
//
//
void setup()
{
    Serial.begin(9600);
    lcd.begin (16,2);
    lcd.backlight();                // illumina lo sfondo del display
    lcd.print ("sistema");
    lcd.setCursor (0,1);
    lcd.print ("pronto");
    delay (3000);
    lcd.clear ();
}
//
//
void loop()
{
    momentocorrente = millis ();
    rilevapicco ();                // lancia routine di rilevazione del picco sulla resistenza
    if ((momentocorrente - momentoesposizione) > 1000) // se e' passato un secondo dall'ultimo display
    {
        momentoesposizione = momentocorrente;
        tensione = (picco * 5.0) / 1024.0; // trasforma in tensione il valore di picco
        picco = 0;                    // azzerà il picco, per consentire una nuova misurazione
        intensita = (tensione / 200.0); // legge di ohm per calcolare l'intensita della corrente
    }
}

```

## Arduino: Sensore di corrente – Current sensor

```
// prodotta dal secondario(200 sono gli ohm della resistenza sulla quale viene misurata la tensione)
intensitarms = intensita * 0.707; // calcola il "valore efficace" o RMS, con la costante 0,707
ampere = intensitarms * 1000; // calcola gli ampere in erogazione sull'utenza (il rapporto
// tra primario e secondario e' 1:1000)
lcd.setCursor (0, 0); // posiziona il cursore sul carattere zero della riga zero
lcd.print ("consumo: ");
lcd.setCursor (0, 1);
lcd.print ("A: ");
lcd.print (ampere);
lcd.print (" W: ");
watt = ampere * 220;
lcd.print (watt);
lcd.print (" ");
}
}
```